

Chapter 1

Client/Server Databases and the Oracle 10g Relational Database

At a Glance

Instructor's Manual Table of Contents

- Overview
- Objectives
- Instructor Notes
- Quick Quizzes
- Discussion Questions
- Key Terms

Lecture Notes

Overview

Chapter 1 covers basic relational database concepts. The chapter begins with a discussion of flat-file storage for data, and then introduces relational databases as a solution to many of the problems with flat files. Database concepts including tables, rows, columns, and keys are covered. A brief introduction to database design discusses entity-relationship modeling and normalization. Students will learn to normalize tables up to third normal form. Database systems can be divided into personal and client/server databases. This book will focus on Oracle 10g, which is a client/server database. Finally, case projects that will serve as examples throughout the text are introduced.

Objectives

- Develop an understanding of the purpose of database systems
- Describe the purpose and contents of an entity-relationship model
- Explain the steps in the normalization process
- Describe the differences between personal and client/server databases
- Understand the Northwoods University student registration database and the Clearwater Traders sales order database

Instructor Notes

Database Systems

Early computer systems stored data in files. These files were made up of fields, or columns, containing specific characteristics and records. A record is a group of related fields that describes a particular entity. Figure 1-1 shows an example data file containing fields and records describing a student. There are numerous problems with the file-based data storage approach, including the fact that a separate program is required to work with each file, and that files may contain duplicate information that can lead to inconsistencies. Databases were developed to solve these problems. In a database, data is stored in a centralized location and is accessed via a database management system (DBMS) and maintained by a database administrator (DBA). Figure 1-4 shows the database approach to data processing.

**Teaching
Tip**

More information on the history of relational databases can be found at:
http://en.wikipedia.org/wiki/Relational_database_management_system

Overview of Relational Databases

In a relational database, data is organized into tables, and each table contains information about a particular entity. The rows in the table contain all of the information about a single entity, or instance, while the columns represent the attributes of the entity. Figure 1-5 contains examples of relational database tables. Database tables are connected to each other through relationships, which are established using key fields. The five main types of key fields are discussed in the following sections.

Primary Keys

A primary key serves to uniquely identify a record. Its value must be unique in the table and cannot be NULL. NULL is a special value used in databases to represent the fact that a value has not been set. It is a good idea to choose a primary key that is numeric and whose value does not change.

Candidate Keys

A candidate key is any column that could potentially serve as a primary key.

Surrogate Keys

In many tables, there is no natural choice for a candidate or primary key. In these cases, it is common practice to add a column for the sole purpose of providing a primary key. Surrogate keys are almost always numerical and often numbered in sequence as records are added to the table. Oracle provides sequences, which can be used to automatically generate numeric values to use as surrogate keys.

**Teaching
Tip**

It is better to create a surrogate key for a table than to try to use an unsuitable value as a primary key.

Foreign Keys

Foreign keys are used to establish relationships between tables. A foreign key in one table contains the value of a primary key in a related table. The foreign key value must match the

related primary key exactly. Figure 1-8 shows a foreign key relationship between the STUDENT and FACULTY tables.

Composite Keys

Composite keys are primary keys that are composed of more than one column. Figure 1-9 shows an example of a composite key.

Quick Quiz

1. In a relational database, data is organized into ____.
Answer: tables
2. True or False: Primary keys are used to form relationships between tables.
Answer: False
3. A primary key composed of more than one column is called a(n) ____ key.
Answer: composite

Database Design

Designing a database ranges from simple to complex depending on the objects involved. The two main tasks involved in designing a database are creating an entity-relationship model and normalizing the database tables.

Entity-Relationship Model

An entity-relationship (ER) model identifies the entities and relationships that will be included in the database. Entities are depicted as boxes, and the relationships between them are depicted using lines. There are three types of relationships that can be formed between entities: one-to-one, one-to-many, and many-to-many. As the names imply, these relationship types are determined by the number of instances of each entity that exist in the relationship. The diagrams for the relationship types are shown in Figures 1-10 through 1-12. One-to-one relationships are rare in relational databases. One-to-many relationships are more commonly used; they are formed by setting the foreign key value in one table to the primary key in another. Many-to-many relationships cannot be represented in the physical database, so it is necessary to introduce a linking table, which holds foreign keys to the two or more related tables.

Normalization

Normalization is used to eliminate redundancy in database tables. It is a step-by-step process that converts unnormalized data to a normalized form. There are several forms, or levels, of normalization. To convert a table to first normal form (1NF), a database designer must remove repeating groups and identify a primary key field. Figure 1-13 shows a database table

containing repeating groups. In Figure 1-14, the repeating groups have been removed from the table.

The second normal form (2NF) applies to tables that are in 1NF and have no partial dependencies. A partial dependency can occur when a composite key is used. If one part of the composite key can be used to identify any other field in the table, a partial dependency exists. In order to remove partial dependencies, it is usually necessary to break the table into two or more tables, each containing data about a specific entity. Pages 14–15 outline a technique that can be used to identify which fields should be placed in each new table.

To reach third normal form (3NF), a table must be in 2NF and then have transitive dependencies removed. A transitive dependency occurs when a field is dependent on another field that is not a primary key. The procedure for removing transitive dependencies is covered on page 16. Again, the solution is to break the table into two or more related tables.

Teaching Tip

More on database normalization can be found at:
<http://www.devshed.com/index.php?option=content&task=view&id=441>

Quick Quiz

1. What are the three types of entity relationships?
Answer: one-to-one, one-to-many, many-to-many
2. True or False: An unnormalized table must contain repeating groups.
Answer: False
3. A(n) ____ dependency must be removed to convert to 3NF.
Answer: transitive

Database Systems

A database system is composed of the DBMS and database applications—client programs that use the DBMS. Note that a server is a computer that shares its resources with other computers, and a server process is a program running on a server. A client is a computer that makes use of resources located on a server.

Personal Database Management Systems

In a personal database—one of the two main types of databases on the market today—the DBMS and the database applications are integrated and appear to the user as one program. Personal databases are normally used by one person at a time, though they can support multiple users by placing data files on a shared network location. Microsoft Access is an example of a

personal database that can be used to create a multiuser application. Figure 1-21 shows how this is accomplished.

In general, personal databases should not be used for mission-critical applications. They impose a heavy network load when configured for multiple users, and do not generally provide the level of fault-tolerance necessary for large-scale applications.

Client/Server Database Management Systems

In a client/server database, the DBMS is run on a server, while database applications are run on client machines. Figure 1-22 illustrates a client/server database environment. Contrast this with Figure 1-21 to see the difference between a client/server database and a multiuser personal database. Note that client/server databases generate far less network traffic since only a small portion of the database data is sent between the client and server. These databases also provide a higher degree of reliability and error handling than personal databases do.

The Oracle 10g Client/Server Database

Oracle 10g is the latest release of Oracle's popular client/server database. Communication between the DBMS and client database applications is handled by Oracle Net, as shown in Figure 1-23. The Oracle client products that will be covered in this text are SQL*Plus; the Oracle 10g Developer Suite, including Forms Builder and Reports Builder; and the Enterprise Manager.

Teaching Tip	The homepage for Oracle 10g is http://www.oracle.com/database/index.html
-------------------------	--

The Database Cases

Two fictional cases will be used throughout this book to illustrate the proper use of Oracle. The remainder of the chapter introduces these cases and outlines the tables that will make up each database.

The Clearwater Traders Sales Order Database

The Clearwater Traders database is used for online order processing and inventory tracking. The tables used for this case are displayed in Figures 1-24 and 1-25.

The Northwoods University Student Registration Database

The Northwoods University database contains information about student enrollment in a fictional university. The tables used for this case are displayed in Figures 1-26 and 1-27.

Quick Quiz

1. True or False: A personal database can only be used by a single person.
Answer: False
2. True or False: Oracle is a personal database.
Answer: False
3. Oracle ____ is used to communicate between the client and the server.
Answer: Net

Discussion Questions

- What advantages (if any) are there to using data files over a database system?
- How do you determine which field(s) to select as the primary key?
- How do you determine which type of relationship exists between entities?
- Why is it important to normalize database tables?
- Why would a personal database be used rather than a client/server database?

Key Terms

- **Attribute:** A data field, or column
- **Candidate key:** Any column that could be used as the primary key
- **Client/server database:** A database in which the DBMS server process runs on one workstation, and the database applications run on separate client workstations across the network
- **Composite key:** A unique key that you create by combining two or more columns
- **Database:** Stores all organizational data in a central location
- **Database administrator (DBA):** The person who is responsible for installing, administering, and maintaining the database
- **Database applications:** Provide the interface that allows users to interact with the database
- **Database management system (DBMS):** An application that performs all routine data-handling operations
- **Entity:** An object about which you want to store data
- **First normal form (1NF):** Data that has been organized in such a manner that it has a primary key and no repeating groups
- **Foreign key:** A column in a table that is a primary key in another table; creates a relationship between the two tables
- **Instance:** A row, or record
- **Key fields:** Establish relationships among records in different tables
- **Many-to-many (N:M) relationship:** A relationship in which an instance can occur multiple times in each entity

- **One-to-many (1:M) relationship:** A relationship in which an instance can only appear once in one entity, but one or more times in the other entity
- **One-to-one (1:1) relationship:** A relationship in which each occurrence of a specific entity is found only once in each set of data
- **Personal database:** A database in which the DBMS and database applications run on the same workstation and appear to the user as a single integrated application
- **Primary key:** A column in a relational database table whose value must be unique for each row
- **Relational database:** A database that stores data in a tabular format
- **Relationships:** Links that show how different records are related
- **Second normal form (2NF):** Data that is in 1NF and contains no partial dependencies
- **Sequences:** Sequential lists of numbers the database automatically generates and that guarantee each primary key value is unique
- **Surrogate key:** A column that you create to be the record's primary key identifier
- **Third normal form (3NF):** Data that is in 2NF and contains no transitive dependencies
- **Unnormalized data:** Data that does not have a primary key identified and/or contains repeating groups